AJ_Tools_UnitTest

User Manual

Version Control	Date	Commentaire (Changement)	Auteur
1.0	18.07.2019	Initial Version	Gabriel Inzirillo

1 Introduction

1.1 What is AJ_Tools_UnitTest

AJ_Tools_UnitTest is a component developed with 4D v17 R5. Its purpose is to give 4D developers a way to do unit test and partial integration tests within 4D.

It is made to be use in development environment but not in production. You can write unit tests and partial integration tests (no UI). You cannot write functional tests with this component.

1.2 Credits

This component was created under the inspiration of the Javascript unit test library RITEway (https://github.com/ericelliott/riteway) developed by Eric Elliot, which is a master in the Test Driven Development.

Those 2 articles was the main resources that inspired me to developer AJ_Tools_UnitTest component.

https://www.sitepoint.com/javascript-testing-unit-functional-integration/

https://medium.com/javascript-scene/rethinking-unit-test-assertions-55f59358253f

2 Getting Started

2.1 Create a new test method

To create a new test, you must create a new method. You are free to name the method as you want.
Here is how the component will recognize a unit test method : The first line must start with

"// __UNIT_TEST"

After, you can create a new test object by calling the New AJ_Tools_UT_describe method :

$test:=New AJ_Tools_UT_describe ("AJ_Tools_UT_assert";Current method name;"AJ_Tools_UT Tests")

Warning : only one "New AJ_Tools_UT_describe" is possible per method !

You can then set the 4 mandatory properties and call the assert member function.

Here this example we test the zz_max method that will return the maximum value passed as parameters :

```
    // __UNIT_TEST

$test:=New UnitTest_Describe ("Test the zz_max method";Current method name;"Math")

$test.given:="no arguments"
$test.should:="return 0"
$test.expected:=0
$test.actual:=zz_max
$test.assert()

$test.given:="1 and 1"
$test.should:="return 1"
$test.expected:=1
$test.actual:=zz_max (1;1)
$test.assert()

```

You can add as many assertions as you need to do your tests.

We recommend you to test one method per test method and to not mix different tests in one big test method.

2.2 Macro

The component come with on macro that will help you to create a new test. You can simply type "_ut_new" and it will create you the minimal structure to make a test.

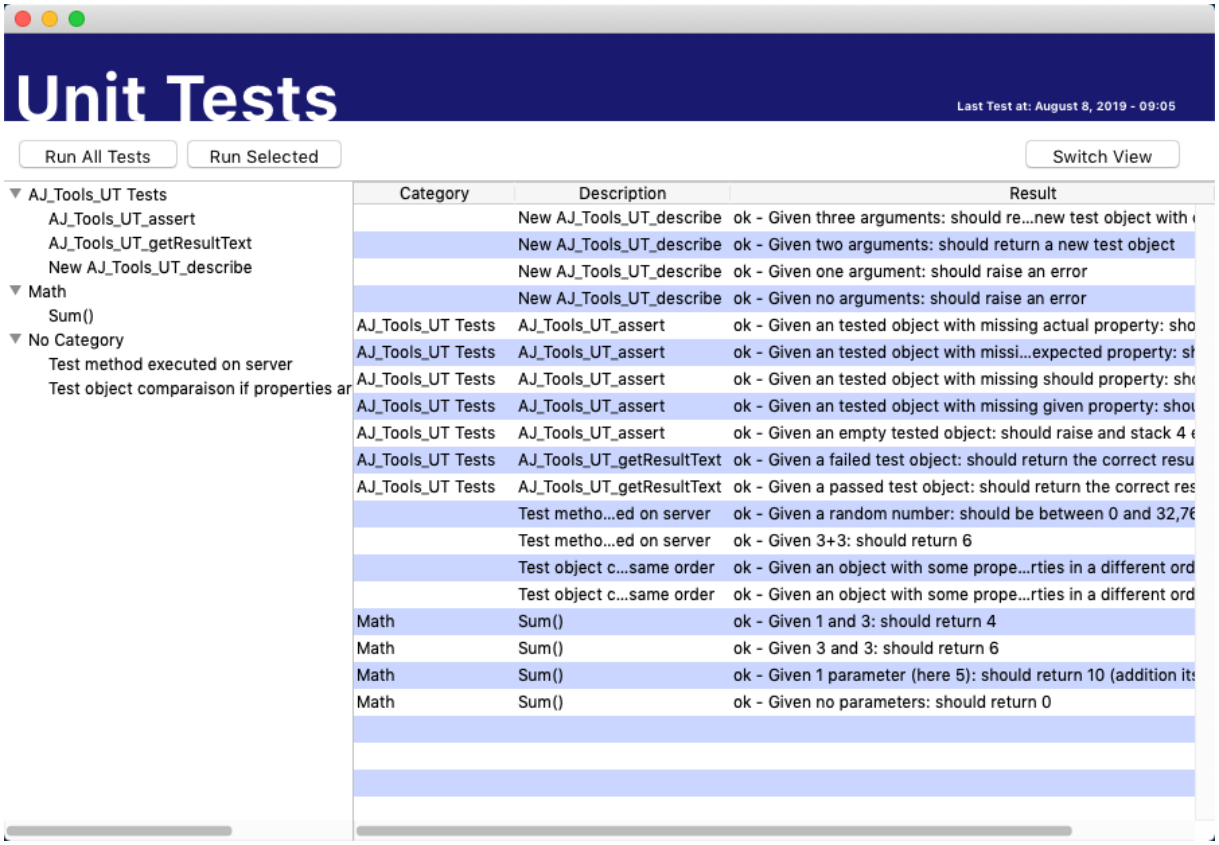
2.3 Main Window

The main window allows you to display and to launch all or partial tests.

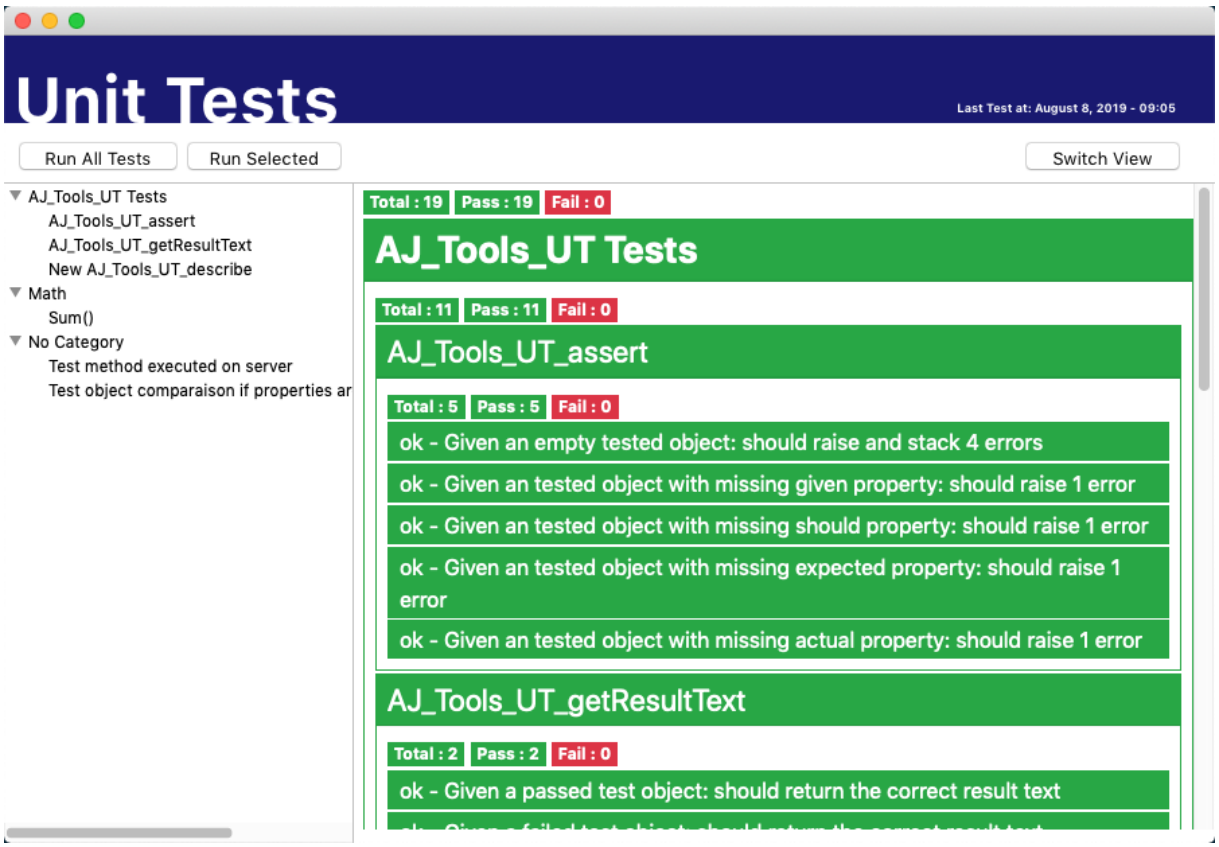
There is 2 way of displaying tests : list view and accordion view.

The list view will be filled with the result while the tests are running and when the test is finish, it will display the accordion view.

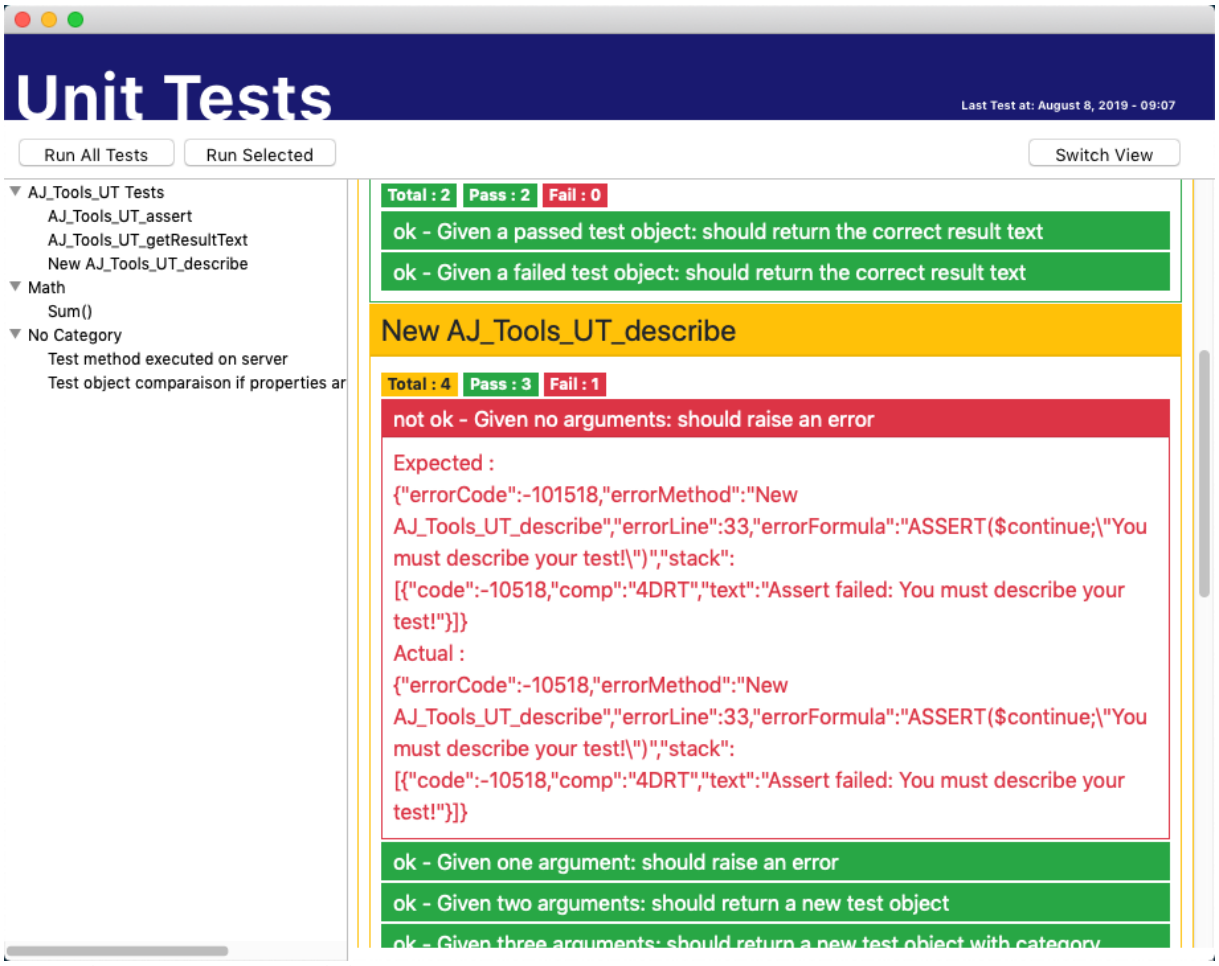
Here are some screenshots :



1 Listbox View



2 Accordion View - All Tests Passed



3 Accordion View Some Test Failed

3 Component methods

3.1 New AJ_Tools_UT_describe

Parameters	• **describe : (text) description of the test** • **method : (text) method that execute the test (must be "Current method name")** • **category : (text) (optional) category of the test. This is used to separate multiple tests in different categories**
Return value	return a unit test object with the assert member function. This object will need 4 parameters (given, should, expected, actual) before to call the assert method
Description	Create a new test. A test can then do multiple assert Only one test must be created by method.

3.2 AJ_Tools_UT_runAll

Parameter(s)	• **methods (collection) (optional) : collections of method to tests**
Return value	(object) parsed results
Description	This method will run all the tests (or only the given methods) and return the results in an object. It will also log the results in a file located at : "<database_path>/Logs/UnitTestsResults.json"

3.3 AJ_Tools_UT_LaunchMainWindow

Parameter(s)	• **none**
Return value	none
Description	This method will launch a window that will list all the categories and descriptions (based on the existing log file). You will be able launch all the tests or only for a selected category/description.

3.4 AJ_Tools_UT_getInfo

Parameter(s)	- **None**
Return value	Version number (text)
Description	This method returns the version number of the component when it has been compiled.

3.5 AJ_Tools_UT_OnErrCall

This method is not made to be used, it gives you some code in the comment section of the method. The code shows you how an error handler should be written to be able to fully test the error handling of your unit tests.

4 Properties List

4.1 Unit Test

Properties of the Unit Test object (all mandatory) :

Name	Type	Description	Default
given	text	Must describe what is the unit under test (a method, a component, whatever…)	null
should	text	Must describe what the test should do	null
expected	any	The expected value	null
actual	any	The actual value that will be compared to the expected to know if the test pass or fail.	null

Unit Test Formula :

Formula name	Parameter(s)	Description
assert	- none	This formula will execute the test. After the execution of the test, all the properties of the unit test object are removed to be ready for the next test.

5 Divers

5.1 Logs

When running the "AJ_Tools_UT_runAll" it will log all the results in a file located at : "<database_path>/Logs/UnitTestsResults.json".

5.2 Run in compiled mode

It is possible to run the tests in compiled mode, however, you must have run the tests once in interpreted mode to be able to run them in compiled mode. This is because we need to parse all the method to know where there are any unit tests (thanks to the first line comment) and this is only possible in interpreted mode. In compile mode it will look at an existing result logs and will be able to know which methods must be launched.

6 Conclusion

The purpose of this document was to introduce you to the theoretical principles of the component as well as the different methods, formulas and properties available to you to manage tests in you application.

As for the practical elements presented, they are intended to allow you to get off to a good start and to understand how the component works.

If you need help to implement the AJ_Tools_UnitTest component in your application. You want to modify or extend its functionalities for a specific use. You want to have the source code of the AJ_Tools_UnitTest component in order to perpetuate its use in your application with future versions of 4D. Feel free to contact us to discuss it.